References

4. Ammann U (1978) Error recovery in recursive descent parsers and run–time storage organiza-
tion, rep. No. 25, Inst. für Informatik der ETH Zürich
ography. Springer, LNCS 323
terdam, rekenafdeling, MR 35. Algol Bulletin, supplement nr. 10

   Functional and Logic Programming, 7th International Symposium, FLOPS 2004, Nara, Japan,
   April 7–9, 2004, Springer, LNCS 2998, pp 196–213
   malization using attribute grammars. Information Processing Letters 7(6):279–284
   (ed) ESOP, Springer, LNCS 788, pp 241–256
   of the AMS 146:29–60
   Addison-Wesley
   Switching and Automata Theory, pp 21–35
30. Jones MP (1995) A system of constructor classes: Overloading and implicit higher-order poly-
    space. In: Proceedings of the 2nd Haskell Workshop
    Lecture Notes in Computer Science, vol 7211, pp 397–416
36. Knuth DE (1965) On the translation of languages from left to right. Information and Control
    8:607–639
    5, pp. 95-96
    6(1):1–5
40. Krieg B (1971) Formal definition of the block concept and some implementation models, mS.
    Thesis, Cornell University
    Vieweg+Teubner
42. Lesk M (1975) Lex – a lexical analyzer generator, cSTR 39, Bell Laboratories, Murray Hill, N.J.


Index

A
acceptor, 15
ADA, 145, 155
ALGOL60, 154
algorithm
  shunting yard, 136
  \W, 165
alphabet, 11, 12
input, 57
alternative, 48
analysis
  data-flow, 7
  lexical, 3, 11
  semantic, 6
  syntactic, 5, 43
analysis phase, 1
anchor terminal, 97
array
  sparse, 34
ASCII, 28
attribute
  dependence, 193, 194
  inherited, 183
  synthesized, 183
attribute equation system, 185
attribute evaluation
  demand-driven, 191
  generation, 191
  parser-directed, 206
attribute grammar, 180
  absolutely noncircular, 200
  L-attributed, 206
  \LL-attributed, 207
  l-ordered, 201
  LR-attributed, 209, 211
  noncircular, 196
  normal form, 183
  ordered, 205
  S-attributed, 209
  semantics, 184
  well-formed, 185, 196
attribute instance, 183
attribute occurrence, 183
automaton
  canonical LR(0), 107
  canonical LR(\textit{k})-automaton
    direct construction, 111
  canonical LR(k), 117
deterministic finite, 16
  finite, 5, 9, 15
  pushdown, 6, 9, 57
axiom, 159

B
back-end, 2
BISON, 181
bottom, 68
bound
  least upper, 68
C
C, 145
class
  declaration, 174
type constructor, 180
COBOL, 145
code generation, 2
code generator, 8
code set, 28
Index

compile time, 6
compiler
  conceptual architecture, 1
directive, 3
  structure of, 1
compiler generation, 10
computation
  accepting, 58
concatenation, 12
$k$, 65
configuration, 17, 58
  error, 124
  final, 17, 58
  initial, 17, 58
conflict
  reduce-reduce, 110, 117
  shift-reduce, 110, 117
constant, 3
constant folding, 7
constant propagation, 7
context-condition, 147
context-free grammar
  ambiguous, 50
  unambiguous, 50
context-free grammar (CFG), 47, 49
C++, 145
C#, 145

D
declaration, 140
  forward, 151
  scope of a, 139
declaredness, 139
dependence
  functional, 183
  production-local, 194
dependence relation
  characteristic
    lower, 197
    production-local, 194
derivable, 48
derivation, 48, 49
  leftmost, 51
  rightmost, 51
deterministic finite automaton (DFA), 16
deterministic pushdown automaton, 58

E
dend vertex, 18
environment
  sort, 175
  type, 158
error
  symptom, 43
  syntactic, 5
error handling
  $LR(k)$-, 124
  $RLL(1)$-, 97
error recovery
  deletion, 127
  insertion, 127
  replacement, 127
evaluation
  short-circuit, 189
expanding transition, 60
expression
  boolean, 189
  regular, 9, 13

F
factorization, 81
final configuration, 58
final state, 16
finite automaton
  characteristic, 103
finite automaton (FA), 15, 16
first, 64
first$_1$-set
  $e$-free, 72
follow, 64
FORTRAN77, 145, 154
front-end, 1
function
  monotonic, 68
future, 79

G
generation
  attribute evaluator, 191
grammar
  attribute, 180
  context-free, 5, 9, 46, 47
    extended, 59
    right-regular, 89
  $LALR(1)$-, 122
  $LL(k)$, 79
  $LL(k)$- (strong), 84
  reduced, 53
  $RLL(1)$-, 92
  underlying, 180

H
handle, 101, 105
HASKELL, 173
hypothesis
  one-error, 126
Index

I
identifier, 3, 139
  applied occurrence of a, 140
  defining occurrence of a, 140
  hidden, 139
  identification, 145, 148
  visible, 139
indentation, 4
initial configuration, 58
initial state, 16, 57
input alphabet, 16, 57
instance declaration, 174
instruction selection, 8
interpretation
  abstract, 7
item
  complete, 59
  context-free, 59
  history, 59
  LR(k)-, 116
  valid, 105
item-pushdown automaton (IPDA), 59, 79

J
JAVA, 145, 155

K
keyword, 3, 38
Kleene star, 13

L
LALR(1), 121, 122
language, 49
  accepted, 17
  regular, 13
lattice
  complete, 68
lexical analysis, 3, 11
LL(k)-
  grammar, 79
  parser (strong), 87
LR(k), 112
LR(k)-item, 116

M
metacharacter, 14
middle-end, 1, 8
monad, 180

N
name space, 140
nonterminal, 47
  left recursive, 85
  productive, 53
  reachable, 55

O
optimization
  machine-independent, 7
overloading, 6, 152, 174
  resolution of, 155

P
panic mode, 97
parenthesis
  nonrecursive, 28
parse tree, 6, 43, 49
parser, 5, 43
  bottom-up, 44, 101
  deterministic, 64
  LALR(1)-, 121
  left-, 64
  LL-, 64
  LR-, 64
  LR(k)-, 102, 117, 118
  Pratt-, 136
  recursive-descent, 92
  right-, 64
  RLL(1)-, 92
  shift-reduce, 101
  SLR(1)-, 121
  top-down, 44
partial order, 68
partition, 26
  stable, 26
PASCAL, 145
polymorphism
  constrained, 173
pragma, 3, 4
prefix, 11, 13
  extendible, 97
  k-, 65
  reliable, 105
  viable, 45, 124
produces, 48
  directly, 48
production rule, 47
PROLOG, 145
pushdown automaton
  deterministic, 58
  item-, 59, 79, 103
  language of a, 58
  with output, 63
pushdown automaton (PDA), 57

Q
qualification, 146
Index

R
reducing transition, 60
reduction
  required, 101
register allocation, 8
regular language, 13
rule, 159
  semantic, 181
run time, 6

S
scanner, 3
  generation, 29
  representation, 34
    compressed, 34
  states, 37
scope, 140
screener, 4, 36
semantic analysis, 6
semantics
  dynamic, 6
  static, 6
sentential form, 49
  left, 51
  right, 51
separator, 3
shifting transition, 60
SLR(1), 121
solution, 163
  most general, 163
sort, 174
sort environment, 175
source program, 3
start symbol, 47
start vertex, 18
state, 57
  actual, 58
  error, 23
  final, 57
inadequate
  LALR(1)-, 122
  SLR(1)-, 122
initial, 57
LR(0)-inadequate, 110
step relation, 17
strategy
  first-fit, 35
string, 28
  pattern matching, 30
strong LL(k)-grammar, 84
subject reduction, 160
subset construction, 21
solution
  idempotent, 163
subword, 13
suffix, 13
symbol, 3, 11, 48
  class, 3
  nonterminal, 47
reserved, 4
start, 47
table, 148, 152
terminal, 47
symbol class, 11, 12
syntactic analysis, 5
syntactic structure, 49
syntax
  abstract, 141
  concrete, 141
syntax analysis
  bottom-up, 101
top-down, 77
syntax error, 43
globally optimal correction, 45
RLL(1)-, 97
syntax tree, 49

T
table
  action-, 117, 118
  goto-, 117
target program, 8
terminal, 47
anchor, 97
transformation phase, 1
transition, 16, 58
  $\epsilon$, 58
  expanding, 60
  reducing, 60
shifting, 60
transition diagram, 17
transition relation, 15, 57
tree
  ordered, 49
type, 140
  cast, 153
class, 173
  consistency, 139
  consistent association, 6
constructor, 180
correctness, 6
environment, 158
judgment, 159
scheme, 169
variable, 160
type inference, 185
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>U</strong></td>
</tr>
<tr>
<td>Unicode, 28</td>
</tr>
<tr>
<td>unification, 163</td>
</tr>
<tr>
<td>union problem</td>
</tr>
<tr>
<td>pure, 74</td>
</tr>
<tr>
<td>unit</td>
</tr>
<tr>
<td>lexical, 3</td>
</tr>
<tr>
<td><strong>V</strong></td>
</tr>
<tr>
<td>validity, 139, 144, 145</td>
</tr>
<tr>
<td>value restriction, 172</td>
</tr>
<tr>
<td>variable</td>
</tr>
<tr>
<td>uninitialized, 139</td>
</tr>
<tr>
<td>variable-dependence graph, 75</td>
</tr>
<tr>
<td>visibility, 139, 144, 147</td>
</tr>
<tr>
<td><strong>W</strong></td>
</tr>
<tr>
<td>word, 12</td>
</tr>
<tr>
<td>ambiguous, 50</td>
</tr>
<tr>
<td><strong>Y</strong></td>
</tr>
<tr>
<td>YACC, 181</td>
</tr>
</tbody>
</table>